Appendix A

Source Code

```
/**
 * Superclass for all classes
 */
import java.io.*;
class Super {
        static FileOutputStream fos;
public static void trace(String s) {
        System.out.println(s);
        }
}

/**
 * Superclass for all record types. Provides methods to read adata in buffer.
 */

import java.io.*;
import java.lang.reflect.*;
import java.lang.Class.*;

class Record extends Super{
        int[] buffer;
        int length;
/**
 * reads data from the sysadata file and fills the objects buffer
 **/
public void fillBuffer(FileInputStream fis, int _length) {
        buffer = new int[_length];

        for (int i= 0; i<_length;i++){
                try{
                        buffer[i] = fis.read();
                }catch(IOException e){
                 trace("Error in "+this.getClass()+":" + e);
                }
         }
}
}

/**
 * SysAdata reads and interprets a COBOL Associated Data (ADATA) file
 * and creates the XMI document.
 */

import java.io.*;
import java.util.Vector;
import com.ibm.ims.message.*;

public class SysAdata extends Super{
  static FileInputStream fis = null;
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
 static CHSRecord chs_rec;
 static CURecord cu_rec;

 static Vector symbolVector = new Vector();
 static Id model = null;
 static int rootLength = 0;
/**
 * addID adds one or more DataItem object to the model recursevily
 */
public static int addID(Id parent, int symbolNumber){
        SymbolRecord symbol = (SymbolRecord) symbolVector.elementAt(symbolNumber);
        try{
                trace("adding "+symbol.getName());
                Id id = parent.add(Type.DATA_ITEM, null);
                id.set(Property.NAME, symbol.getName());
                id.set(Property.TYPE, symbol.getAttribute());
                id.set(Property.LENGTH, symbol.getLength());
                id.set(Property.HAS_VALUE, ""+symbol.hasValue());
                if(symbol.hasValue()){
                        rootLength = rootLength + symbol.getSize();
                        id.set(Property.VALUE, "");
                }

                symbolNumber++;
                if((symbol.hasValue() == false) && (symbolNumber < symbolVector.size())){
                        SymbolRecord nextSymbol = (SymbolRecord)
symbolVector.elementAt(symbolNumber);
                        if(nextSymbol.redefines()) {
                                symbolNumber = addID(id,symbolNumber);
                        }
                        else {
                                while((symbol.getLevel()<nextSymbol.getLevel()) && (symbolNumber
< symbolVector.size())){
                                        symbolNumber = addID(id,symbolNumber);
                                        if(symbolNumber < symbolVector.size())
                                                nextSymbol = (SymbolRecord)
symbolVector.elementAt(symbolNumber);
                                }
                        }
                }
        }catch(Exception e){e.printStackTrace();}

        // returns the number of the next symbol to be processed
        return symbolNumber;
}
public static void main(String args[]){
        boolean loopflag = true;
        int counter = 0;

if(args.length > 1){
        //open adata file which is given as first argument in the parameter list
        try {
                fis = new FileInputStream(args[0]);
                //fis = new FileInputStream("d:/jan/parser/sysadata/data.adt");
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
                }
        catch(java.io.FileNotFoundException e){
                System.out.println("error" + e);
                return;
        }

        //reads first common header section (CHSRecord) from file
        chs_rec = new CHSRecord(fis);

/**
 * loop reads records from the adata file
 * checks for CURecord, SymbolRecord and skips all other records
 * stores SymbolRecords in symbolVector
 */
        do{
                switch (chs_rec.getrecordType()){
                        case 2: {
                                //Compilation Unit Record, 02hex
                                cu_rec = new CURecord(fis);
                                if(cu_rec.getType() == 1)
                                  loopflag = false;
                                chs_rec.fill(fis);
                                break;}
                        case 66:{
                                //Symbol Record, 042hex
                                   SymbolRecord symbolrecord = new
SymbolRecord(fis,chs_rec.getrecordLength());
                                if(symbolrecord.getType()==64)
                                        symbolVector.addElement(symbolrecord);
                                chs_rec.fill(fis);
                                break;}
                        default:{
                                //all other records are skipped
                                skip(chs_rec.getrecordLength());
                                chs_rec.fill(fis);
                                break;}
                }
        }while(loopflag);

        try{
                fis.close();
        }catch(IOException e){trace(""+e);}

        try {
                model = Model.instance().getSession();
                Id root = model.add(Type.DATA_ITEM, null);
                root.set(Property.NAME, "ROOT");
                root.set(Property.TYPE, "ROOT");
                root.set(Property.HAS_VALUE, "False");

                do{
                        counter = addID(root, counter);
                }while(counter<symbolVector.size());
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
                root.set(Property.LENGTH, ""+rootLength);
                Model.instance().save(root, args[1], Model.DEFAULT, new java.util.Vector());
                //Model.instance().save(root, "sample.xml", Model.DEFAULT, new java.util.Vector());
        }catch (Exception e) {e.printStackTrace();}
}
else
        trace("No input and output file given\nSyntax is: java SysAdata input.adt output.xml");
}
/**
 * skips the given amount of bytes in the SysAdata input file.
 */
public static void skip(int length) {
        try{
                fis.skip(length);
                }catch(java.io.IOException e){trace(""+e);}
}
}

/**
 * CHSRecord: Common header Section - x0001
 * 12byte long, common for all record types
 */

import java.io.*;
class CHSRecord extends Record{

  public  byte  langCode;    // Language Code
  public  short recordType;  // Record Type
  public  byte  sysadataLevel;// SysAdata Architecture Level
  public  byte  contFlag;    // bit 1: record is continued; bit 2: integers are Little-Endian; bits 3-8: reserved
  public  byte  editionLevel; // Indicates a new format for a specific record type;  usually 0
  public  int   reserve;     // Reserved for future use
  public  short recordLength; // Record Length following header (in bytes)

/**
 * Constructor
 **/

public CHSRecord(FileInputStream fis){

        fill(fis);
  }
/**
 * fills all fields of the Common Header Section Record
 */
public void fill(FileInputStream fis) {

        super.fillBuffer(fis,12);
        langCode = new Integer(buffer[0]).byteValue();
        recordType = new Integer(buffer[1]+(buffer[2]*256)).shortValue(); //xchanges byte 2 and 3; shifts
byte3 left
        sysadataLevel = new Integer(buffer[3]).byteValue();
        contFlag = new Integer(buffer[4]).byteValue();
        editionLevel = new Integer(buffer[5]).byteValue();
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
        recordLength = new Integer(buffer[10]+(buffer[11]*256)).shortValue();
}
/**
 * returns length of the following record
 */

public short getrecordLength(){
        return(recordLength);
  }
/**
 * returns type of the following record in decimal
 */
public short getrecordType(){
        return(recordType);
  }
/**
 * prints out attributes of CHSRecord
 */
public void print(){
        trace("langCode " + langCode + ";recordType " +recordType+  ";sysadataLevel "+sysadataLevel+
";contFlag "+contFlag+ ";editionLevel "+editionLevel+ ";recordLength "+recordLength);
  }
}

/**
 * Compilation Unit Start/End Record - x0002
 * 8 bytes long
 */
import java.io.*;

class CURecord extends Record{
        short type;
/**
 * CURecord constructor
 */
public CURecord(FileInputStream fis) {
        fill(fis);
}
/**
 * fills all needes fields of the Compilation Unit Start/End Record
 */
public void fill(FileInputStream fis) {
        super.fillBuffer(fis,8);

        type = new Integer(buffer[0]+(buffer[1]*256)).shortValue();
}
/**
 * returns the type of the CURecord
 * compilation unit types are: x0000=Start, x0001=end
 */
public int getType() {
        return type;
}
}
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
/**
 * Symb lRecord: Common header Section - x0001
 * variable lenght, contains description of all symbols
 *
 * All get methods are used to fill the DataItems in the XMI Document.
 */

import java.math.*;
class SymbolRecord extends Record {
        //fields of the Common Header Section
        int symbolId;
        byte  level;
        byte  symbolType;
        byte  symbolAttribute;
        byte[]  clauses = new byte[1];
        BigInteger clausesB;
        byte[]  flags1 = new byte[1];
        BigInteger flags1B;
        int   size;
        int parentId;
        int redefinedId;
        short  symbolNameLen;
        String symbolName;

        //symbolAttributeValues are put into the attribute tag of the DataItem
        static String symbolAttributeValues[] = {
                        "",
                        "Numeric",
                        "Alphanumeric",
                        "Group",
                        "Pointer",
                        "IndexDataItem",
                        "IndexName",
                        "Condition",
                        "","","","","","","","",
                        "File",
                        "SortFile",
                        "","","","","",
                        "ClassName",
                        "ObjectReference"};

/**
 * Constructor
 */
public SymbolRecord(java.io.FileInputStream fis, int length) {
        fill(fis, length);
}
/**
 * fills needed fields of Symbol Record
 */
public void fill(java.io.FileInputStream fis, int length) {
        super.fillBuffer(fis,length);
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
        symbolId = new Integer(buffer[0] + buffer[1]*16*16 + buffer[2]*16*16*16*16 +
buffer[3]*16*16*16*16*16*16).intValue();
        level = new Integer(buffer[8]).byteValue();
        symbolType = new Integer(buffer[10]).byteValue();
        symbolAttribute = new Integer(buffer[11]).byteValue();
        clauses[0] = new Integer(buffer[12]).byteValue();
        clausesB = new BigInteger(clauses);
        flags1[0] = new Integer(buffer[13]).byteValue();
        flags1B = new BigInteger(flags1);
        size = new Integer(buffer[20] + buffer[21]*16*16 + buffer[22]*16*16*16*16 +
buffer[23]*16*16*16*16*16*16).intValue();
        parentId = new Integer(buffer[44] + buffer[45]*16*16 + buffer[46]*16*16*16*16 +
buffer[47]*16*16*16*16*16*16).intValue();
        redefinedId = new Integer(buffer[48] + buffer[49]*16*16 + buffer[50]*16*16*16*16 +
buffer[51]*16*16*16*16*16*16).intValue();
        symbolNameLen = new Integer(buffer[90] + buffer[91]*16*16).shortValue();

        char[] temp = new char[symbolNameLen];
        int j=0;
        for(int i=104;i<(104+symbolNameLen);i++){
                temp[j]=(char)buffer[i];
                j++;
        }
        symbolName = new String(temp);

}
public String getAttribute(){
                return symbolAttributeValues[symbolAttribute];
        }
public String getLength(){
                return new Integer(size).toString() ;
        }
public int getLevel(){
                return level;
}
public String getName(){
                return symbolName;
}
public int getParentId(){
                return parentId;
}
public int getSize(){
                return size;
}
public int getSymbolId(){
                return symbolId;
}
public int getType(){
        return symbolType;
}
/**
 * bit 7 is 1 if the symbol is redefined
 * symbol is group if symbolAttribute == 3
 */
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
public boolean hasValue(){
        if(flags1B.testBit(7) || symbolAttribute == 3)
                return false;
        else
                return true;
}
public void print() {
        trace("symbolID:"+ symbolId+ ", level:"+ level+", parentId:"+parentId+", size:"+size+",
redefined:"+flags1B.testBit(7)+", symbolAttribute:"+symbolAttribute+", redefinedId:"+redefinedId+",
name:"+symbolName+" ,hasValue:"+hasValue());
}
/**
 * bit 5 is high if this symbol redefines another one
 */

public boolean redefines(){
                return clausesB.testBit(5);
        }
}
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Appendix B

Classes

In various embodiments, each record type is implemented as a class that understands how to handle the binary data from the SysAdata file 78 and provide the data to other classes via defined access methods. Figure 9 illustrates the relationship among the classes. A more detailed description on attributes and methods of each class follows. The source code of all classes may be found in Appendix A.

Class Super

This class offers methods and attributes that are needed by all classes.

Attributes

No Attributes defined.

Methods

- public static void trace(String s)

  trace is used to trace error statements. It replaces Java's System.out.print().

Class Record

This class offers methods and attributes that are needed by all Record classes.

Attributes

- int[] buffer

  Array of ints that stores the actual data from the SysAdata file 78.

- int length

  Stores the size of the buffer array.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Methods

- public void fillBuffer(FileInputStream fis, int _length)

  Fills the buffer array.

Class SysAdata

SysAdata wraps methods and attributes to read the SysAdata file 78 and create the XML document template 58. Initially, the complete SysAdata file 78 is processed and all data-entry symbols are saved in the symbolVector. Thereafter, the symbolVector is processed and an XMI model that resembles the document is created.

Attributes

- FileInputStream fis

  Wraps the input adata file.

- CHSRecord chs_rec

  Is used to temporarly save a CHSRecord.

- CURecord cu_rec

  Is used to temporarly save a CURecord.

- Vector symbolVector

  The symbolVector stores all SymbolRecords that are extracted from the adata file.

- Id model

  model is the root element for the XMI toolkit object hierarchy.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

- int rootLength

  Stores the added up length of all DataItem.VALUEs.

Methods

- public static void main(String args[])

  The main method contains the actual loop, in which the SysAdata file is processed. The assembly of the model for the XML document template 58 is initiated here as well.

- public static int addID(Id parent, int symbolNumber)

  Recursive method to assemble the model that is then saved as XML. Parameters are:

  - the parents Id, such that the next symbol can be added on the right level.

  - the position number of the symbol to be added in the symbolVector.

  addID returns the index of the next SymbolRecord object to be processed.

- public static void skip(int length)

  Replaces FileInputStream.skip().

Class CHSRecord

Throughout the processing of the SysAdata file 78, each common header section instantiates this class.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Attributes

A description of these attributes is found in Table 1.

- byte langCode
- short recordType
- byte sysadataLevel
- byte contFlag
- byte editionLevel
- public int reserve
- public short recordLength

Methods

- public CHSRecord(FileInputStream fis)

  The constructor instantiates the CHSRecord class. The FileInputStream object is handed over from the calling instance.

- public void fill(FileInputStream fis)

  Calls Record.fillBuffer() and then extracts the binary data stored in buffer[] into the referring attribute.

- public short getrecordLength()

  Returns the length of the following record. Is needed in one embodiment to process the input file correctly.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

- public short getrecordType()

  Returns the record type.

Class CURecord

A compilation unit start/end record instantiates this class. The CURecord class is used to control the processing of the SysAdata file 78. In certain embodiments, if getType() returns 1, the processing is stopped.

Attributes

- short type

Methods

- public CURecord(FileInputStream fis)

  The constructor instantiates the CURecord class. The FileInputStream object is handed over from the calling instance.

- public void fill(FileInputStream fis)

  Calls Record.fillBuffer() and then extracts the binary data stored in buffer[] into the refering attribute.

- public int getType()

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Class SymbolRecord

Any symbol record found during the processing of the SysAdata file 78 instantiates this class.

Attributes

A detailed description of these attribute is given in Table 3. The clauses and the flags1 attribute are converted into BigIntegers and are therefore stored in an array with size one to provide the correct input to the BigInteger constructor. The BigInteger object is used, because it allows bit-operations on its value.

- int symbolId
- byte level
- byte symbolType
- byte symbolAttribute
- byte[] clauses = new byte[1]
- BigInteger clausesB
- byte[] flags1 = new byte[1]
- BigInteger flags1B
- int size
- int parentId
- int redefinedId
- short symbolNameLen

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

- String symbolName
- static String symbolAttributeValues[]

This array contains descriptive names for the DataItem.TYPE tag.

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

Appendix C

Document Type Definition (DTD)

```
<?xml version="1.0" encoding="UTF-8" ?>

<!-- XMI Automatic DTD Generation      -->
<!-- Metamodel: DataItem               -->

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI is the top-level XML element for XMI transfer text -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI (XMI.header, XMI.content?, XMI.difference*,
        XMI.extensions*) >
<!ATTLIST XMI
        xmi.version CDATA #FIXED "1.0"
        timestamp CDATA #IMPLIED
        verified (true | false) #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.header contains documentation and identifies the model, -->
<!-- metamodel, and metametamodel                         -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.header (XMI.documentation?, XMI.model*, XMI.metamodel*,
                XMI.metametamodel*) >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- documentation for transfer data                      -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.documentation (#PCDATA | XMI.owner | XMI.contact |
                XMI.longDescription | XMI.shortDescription |
                XMI.exporter | XMI.exporterVersion |
                XMI.notice)* >

<!ELEMENT XMI.owner ANY >

<!ELEMENT XMI.contact ANY >

<!ELEMENT XMI.longDescription ANY >

<!ELEMENT XMI.shortDescription ANY >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ELEMENT XMI.exporter ANY >

<!ELEMENT XMI.exporterVersion ANY >

<!ELEMENT XMI.exporterID ANY >

<!ELEMENT XMI.notice ANY >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.element.att defines the attributes that each XML element -->
<!-- that corresponds to a metamodel class must have to conform to -->
<!-- the XMI specification.                               -->
<!-- ____________________________________________________ -->

<!ENTITY % XMI.element.att
        'xmi.id ID #IMPLIED xmi.label CDATA #IMPLIED xmi.uuid
         CDATA #IMPLIED ' >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.link.att defines the attributes that each XML element that -->
<!-- corresponds to a metamodel class must have to enable it to -->
<!-- function as a simple XLink as well as refer to model -->
<!-- constructs within the same XMI file.                 -->
<!-- ____________________________________________________ -->

<!ENTITY % XMI.link.att
        'xml:link CDATA #IMPLIED inline (true | false) #IMPLIED
         actuate (show | user) #IMPLIED href CDATA #IMPLIED role
         CDATA #IMPLIED title CDATA #IMPLIED show (embed | replace
         | new) #IMPLIED behavior CDATA #IMPLIED xmi.idref IDREF
         #IMPLIED xmi.uuidref CDATA #IMPLIED' >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.model identifies the model(s) being transferred  -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.model ANY >
<!ATTLIST XMI.model
      %XMI.link.att;
      xmi.name    CDATA #REQUIRED
      xmi.version  CDATA #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.metamodel identifies the metamodel(s) for the transferred -->
<!-- data                                                 -->
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!-- ____________________________________________________ -->

<!ELEMENT XMI.metamodel ANY >
<!ATTLIST XMI.metamodel
        %XMI.link.att;
        xmi.name    CDATA #REQUIRED
        xmi.version  CDATA #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.metametamodel identifies the metametamodel(s) for the -->
<!-- transferred data                                     -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.metametamodel ANY >
<!ATTLIST XMI.metametamodel
        %XMI.link.att;
        xmi.name    CDATA #REQUIRED
        xmi.version  CDATA #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.content is the actual data being transferred     -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.content ANY >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.extensions contains data to transfer that does not conform -->
<!-- to the metamodel(s) in the header                    -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.extensions ANY >
<!ATTLIST XMI.extensions
        xmi.extender CDATA #REQUIRED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- extension contains information related to a specific model -->
<!-- construct that is not defined in the metamodel(s) in the header -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.extension ANY >
<!ATTLIST XMI.extension
        %XMI.element.att;
        %XMI.link.att;
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
        xmi.extender   CDATA #REQUIRED
        xmi.extenderID CDATA #REQUIRED
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI.difference holds XML elements representing differences to a -->
<!-- base model                                                   -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.difference (XMI.difference | XMI.delete | XMI.add |
                XMI.replace)* >
<!ATTLIST XMI.difference
        %XMI.element.att;
        %XMI.link.att;
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI.delete represents a deletion from a base model           -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.delete EMPTY >
<!ATTLIST XMI.delete
        %XMI.element.att;
        %XMI.link.att;
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI.add represents an addition to a base model               -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.add ANY >
<!ATTLIST XMI.add
        %XMI.element.att;
        %XMI.link.att;
        xmi.position CDATA "-1"
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI.replace represents the replacement of a model construct  -->
<!-- with another model construct in a base model                 -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.replace ANY >
<!ATTLIST XMI.replace
        %XMI.element.att;
        %XMI.link.att;
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
        xmi.position CDATA "-1"
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- XMI.reference may be used to refer to data types not defined in -->
<!-- the metamodel                                        -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.reference ANY >
<!ATTLIST XMI.reference
        %XMI.link.att;
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- This section contains the declaration of XML elements -->
<!-- representing data types                              -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.TypeDefinitions ANY >

<!ELEMENT XMI.field ANY >

<!ELEMENT XMI.seqItem ANY >

<!ELEMENT XMI.octetStream (#PCDATA) >

<!ELEMENT XMI.unionDiscrim ANY >

<!ELEMENT XMI.enum EMPTY >
<!ATTLIST XMI.enum
        xmi.value CDATA #REQUIRED
>

<!ELEMENT XMI.any ANY >
<!ATTLIST XMI.any
        %XMI.link.att;
        xmi.type CDATA #IMPLIED
        xmi.name CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTypeCode (XMI.CorbaTcAlias | XMI.CorbaTcStruct |
                XMI.CorbaTcSequence | XMI.CorbaTcArray |
                XMI.CorbaTcEnum | XMI.CorbaTcUnion |
                XMI.CorbaTcExcept | XMI.CorbaTcString |
                XMI.CorbaTcWstring | XMI.CorbaTcShort |
                XMI.CorbaTcLong | XMI.CorbaTcUshort |
                XMI.CorbaTcUlong | XMI.CorbaTcFloat |
                XMI.CorbaTcDouble | XMI.CorbaTcBoolean |
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
                    XMI.CorbaTcChar | XMI.CorbaTcWchar |
                    XMI.CorbaTcOctet | XMI.CorbaTcAny |
                    XMI.CorbaTcTypeCode | XMI.CorbaTcPrincipal |
                    XMI.CorbaTcNull | XMI.CorbaTcVoid |
                    XMI.CorbaTcLongLong |
                    XMI.CorbaTcLongDouble) >
<!ATTLIST XMI.CorbaTypeCode
        %XMI.element.att;
>

<!ELEMENT XMI.CorbaTcAlias (XMI.CorbaTypeCode) >
<!ATTLIST XMI.CorbaTcAlias
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcStruct (XMI.CorbaTcField)* >
<!ATTLIST XMI.CorbaTcStruct
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcField (XMI.CorbaTypeCode) >
<!ATTLIST XMI.CorbaTcField
        xmi.tcName CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcSequence (XMI.CorbaTypeCode |
                    XMI.CorbaRecursiveType) >
<!ATTLIST XMI.CorbaTcSequence
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaRecursiveType EMPTY >
<!ATTLIST XMI.CorbaRecursiveType
        xmi.offset CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcArray (XMI.CorbaTypeCode) >
<!ATTLIST XMI.CorbaTcArray
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcObjRef EMPTY >
<!ATTLIST XMI.CorbaTcObjRef
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcEnum (XMI.CorbaTcEnumLabel) >
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ATTLIST XMI.CorbaTcEnum
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcEnumLabel EMPTY >
<!ATTLIST XMI.CorbaTcEnumLabel
        xmi.tcName CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcUnionMbr (XMI.CorbaTypeCode, XMI.any) >
<!ATTLIST XMI.CorbaTcUnionMbr
        xmi.tcName CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcUnion (XMI.CorbaTypeCode, XMI.CorbaTcUnionMbr*) >
<!ATTLIST XMI.CorbaTcUnion
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcExcept (XMI.CorbaTcField)* >
<!ATTLIST XMI.CorbaTcExcept
        xmi.tcName CDATA #REQUIRED
        xmi.tcId   CDATA #IMPLIED
>

<!ELEMENT XMI.CorbaTcString EMPTY >
<!ATTLIST XMI.CorbaTcString
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcWstring EMPTY >
<!ATTLIST XMI.CorbaTcWstring
        xmi.tcLength CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcFixed EMPTY >
<!ATTLIST XMI.CorbaTcFixed
        xmi.tcDigits CDATA #REQUIRED
        xmi.tcScale  CDATA #REQUIRED
>

<!ELEMENT XMI.CorbaTcShort EMPTY >

<!ELEMENT XMI.CorbaTcLong EMPTY >

<!ELEMENT XMI.CorbaTcUshort EMPTY >

<!ELEMENT XMI.CorbaTcUlong EMPTY >
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ELEMENT XMI.CorbaTcFloat EMPTY >
<!ELEMENT XMI.CorbaTcDouble EMPTY >
<!ELEMENT XMI.CorbaTcBoolean EMPTY >
<!ELEMENT XMI.CorbaTcChar EMPTY >
<!ELEMENT XMI.CorbaTcWchar EMPTY >
<!ELEMENT XMI.CorbaTcOctet EMPTY >
<!ELEMENT XMI.CorbaTcAny EMPTY >
<!ELEMENT XMI.CorbaTcTypeCode EMPTY >
<!ELEMENT XMI.CorbaTcPrincipal EMPTY >
<!ELEMENT XMI.CorbaTcNull EMPTY >
<!ELEMENT XMI.CorbaTcVoid EMPTY >
<!ELEMENT XMI.CorbaTcLongLong EMPTY >
<!ELEMENT XMI.CorbaTcLongDouble EMPTY >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- METAMODEL:  DataItem                                 -->
<!-- ____________________________________________________ -->


<!ELEMENT DataItem.child (DataItem)* >

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- METAMODEL CLASS:  DataItem                           -->
<!-- ____________________________________________________ -->


<!ELEMENT DataItem.Name (#PCDATA | XMI.reference)* >

<!ELEMENT DataItem.Type EMPTY >
<!ATTLIST DataItem.Type
        xmi.value ( Root | Numeric | Alphanumeric | Group ) #REQUIRED
>

<!ELEMENT DataItem.Length (#PCDATA | XMI.reference)* >

<!ELEMENT DataItem.hasValue EMPTY >
```


MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

```
<!ATTLIST DataItem.hasValue
        xmi.value ( true | false ) #REQUIRED
>

<!ELEMENT DataItem.Value (#PCDATA | XMI.reference)* >

<!ELEMENT DataItem.parent (DataItem)? >

<!ELEMENT DataItem (DataItem.Name?, DataItem.Type?, DataItem.Length?,
            DataItem.hasValue?, DataItem.Value?, XMI.extension*,
            DataItem.parent?, DataItem.child*)? >
<!ATTLIST DataItem
        %XMI.element.att;
        %XMI.link.att;
>
```

MADSON & METCALF, P.C.
ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101